International Journal of Computer Networking,
Wireless and Mobile Communications (IJCNWMC)
ISSN(P): 2250-1568; ISSN(E): 2278-9448
Vol. 3, Issue 5, Dec 2013, 85-92
© TJPRC Pvt. Ltd.

ISDRP: AN IMPROVED SECURE AND DISTRIBUTED PROTOCOL FOR WIRELESS SENSOR NETWORK

CHHABEEL KAUR[1] & HARDEEP SINGH[2]

[1]Chandigarh Engineering College, Landran, Mohali Punjab, India

[2]Chandigarh Group of Colleges-Coe, Landran, Mohali Punjab, India

ABSTRACT

A Reprogramming service should be efficient, reliable and secured in Wireless sensor network. Wireless reprogramming for wireless sensor network emphasize over the process of changing or improving the functionality of simulation or existing code. For challenging and on demand security purpose, a secure and distributed routing protocol ISDRP is developed to improve the security and broadcasting nature of the sensor network. This paper proposes a reprogramming protocol, which is based on hierarchy of energies in network. The proposed protocol follows the identity-based cryptography using public key and private key to secure the reprogramming and storage requirements of each node. Moreover, this work demonstrates the security concepts, which deals over the key encryption properties using heap sort algorithm. The keys are distributed to the network as per the sorting and communication capabilities to improve the broadcast or communication nature of the network. Furthermore, the confidentiality parameter is enhanced by changing the private key values after certain interval of time for cluster head in respect to different public keys. The result of proposed work shows high efficiency rate, which is clear with the throughput results.

KEYWORDS: Wireless Sensor Network, Reprogramming, Zero Knowledge Proof, Heapsort Algorithm, Identity Based Cryptography

INTRODUCTION

A sensor network [8] is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. Each node consists of a microcontroller (performs tasks, processes data and control components), transceiver (combined functionality of transmitter and receiver), external memory (on chip or off chip), power source (rechargeable and non- rechargeable batteries) and one or more sensor.Sensor networks have been proposed for a wide variety of application areas such as Area monitoring, intrusion detection and tracking smart home monitoring and many more.

As these applications makes nodes to physical interact with them there is need of maintenance. Thus users must be able to add or change the functionality of a deployed network to fully utilize its capabilities. It is clear that network reprogramming [4] is required for the success of wireless sensor network. The two different methods of reprogramming are first by system administrator called code dissemination and other by individual sensors from network on demand called code acquisition.

All of the previous protocols such as Deluge [2], Seluge [7] and SDRP [3] have high propagation delays and due to this overhead problem. Proposed scheme is easy to understand and decrease the propagation delay enabling high throughput rates. Thus energy overhead is also lesser comparatively. We have implemented this on ns 2 simulator with Linux operating system.

RELATED WORK

Initially we discussed about several recent works about different proposed schemes on secured code dissemination for wireless networks.

Adam Chlipala represents that Deluge [2] as a reliable data dissemination protocol for propagating large amounts of data from one or more source nodes to other nodes over a multihop in a network. It does above while maintaining a constant amount of local state. Thus it demonstrates the energy required to distribute this data is within the allowable per-mote energy budgets. Many optimizations on performance of Deluge are also examined. Sangwon Hyun presents a secure extension of Deluge, an open source state of-the-art code dissemination for wireless sensor network. It is efficient, secure, robust and DOS-resistant. It also includes integrity protection of code images and immunity from all DOS attacks.

The experimentation evaluation shows efficiency of Seluge[7] in practice on micaz motes. Patrick E. Lanigan presents a new protocol Sluice[3], which is an extension aiming for the progressive, resource-sensitive verification of updates within sensor networks by exploiting a single digital signature per update, along with a hash-chain construction over pages of the update. It provides enhanced security preventing malicious nodes from propagating or installing malicious updates on uncompromised nodes within the system.

Based on hierarchy Shirshu Varma [10] represent protocol for reprogramming in which we have divided the nodes as super nodes (cluster heads) and normal nodes. We first send codes to nodes in the upper layer of the node hierarchy (i.e., super nodes). Then super nodes reprogram other nodes in their local areas. With these approach problems like sender redundancy, data redundancy in classic flooding is overcomed.

All existing protocols are based on centralized approach. Daojing He represents a new distributed based protocol called SDRP [9] where there exist multiple authorized network users to simultaneously and directly reprogram sensor nodes without involving base station. The protocol uses identity-based cryptography [1] for secure reprogramming. It is a cryptography scheme in which the public key is the identity key is the identity of the user instead of some random generated number. Any string can be used as the public key, as long as it undeniably identifies the user. It has few demerits such as no support given to confidentiality as in some applications data is to be kept confidential due to the possibility of message interception. Also propagation delay is more. When sensor node’s radio is always on during the reprogramming process, energy consumption of the node depends chiefly on the completion time (i.e. propagation delay). There is energy overhead in SDRP similar to that of Deluge Seluge.

IMPROVED PROTOCOL

Heapsort Algorithm

Suppose H is a complex binary tree with ‘n’ elements. The ‘H’ is called heap if it has following property: The value at N (each node) is greater then or equal to the value at each of the children of N.Heapsort [5] is a comparison based sorting algorithm to create a sorted array (or list). Heapsort consists of mainly two phases, in first phase it is building of a heap H out of elements of A (array) and, in second repeatedly deletion of the root element takes place of H. Since the root H always contains the largest node in H, second phase deletes the elements of A in Decreasing order.It has worst-case runtime complexity as$O(n \log n)$, which is its advantage.

Identity Based Cryptography for Security

It is a cryptographic scheme in which the public key is the identity of the (user e.g. his e-mail address) instead of same random generated number. The obvious advantage of this is that it eliminated the need for users to look up public keys in a directory and the use of certificate binding the public key to an identity. Here a third party called generator center or Private key generator (PKG) also exists. To operate PKG publishes a master public and a corresponding master private key. Given the master public key, any party can compute a public key corresponding the identity ID by combining master public value with the identity value. To obtain the corresponding private keys the party authorized to use the identity ID contact with PKG, which used the master private key to generate the private key for ID. As a result, parties a may encrypt messages with no prior distribution of keys between participants.

An identity-based encryption scheme [1] E is specified by four randomized algorithms: Setup, Extract, Encrypt, Decrypt:

- **Setup:** takes a security parameter k and returns params (system parameters) and master-key. The system parameters include a description of a finite message space M, and a description of a finite cipher text space C. Intuitively, the system parameters will be publicly known, while the master key will be known only to the "Private Key Generator" (PKG).
- **Extract:** takes as input params, master key, and an arbitrary$ID \in \{0,1\}^*$, and returns a private key d. Here ID is an arbitrary string that will be used as a public key, and d is the corresponding private decryption key. The Extract algorithm extracts a private key from the given public key.
- **Encrypt**: takes as input params, ID, and $M \in M$. It returns a ciphertext $C \in C$.
- **Decrypt:** takes as input params, $C \in C$, and a private key d. It returns$M \in M$.

These algorithms must satisfy the standard consistency constraint, namely when d is the private key generated by algorithm Extract when it is given ID as the public key, then:

$$\forall\, M \in M: Decrypt(params, C, d) = M \text{ where } C = Encrypt(params, ID, M)$$

Zero Knowledge Proofs for Confidentiality

A zero-knowledge proof (ZKP) [11] is a proof of some statement, which reveals nothing other than the veracity of the statement i.e. no additional information conveyed.

A zero-knowledge proof must satisfy three properties [12]:

- **Completeness:** if the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by an honest prover.
- **Soundness:** if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability
- **Zero-Knowledge:** if the statement is true, no cheating verifier learns anything other than this fact. This is formalized by showing that every cheating verifier has some simulator that, given only the statement to be proven, can produce a transcript that "look like" an interaction between the honest prover and the cheating verifier.

Authentication systems motivated zero knowledge proofs where one party wants to prove its identity to a second party via some secret information such as password. But the other party doesn't know about this secret.It could be explained withgraph coloring example. There is a verifier who can check that any pair of adjacent vertices is colored correctly, that no two adjacent vertices are colored the same, but he cannot unite theinformation and produce the entire coloring of the graphs.

PROPOSED WORK

Since our proposed work is highly based over the reprogramming concepts. So the following screenshots depicts the somehow information for the same.

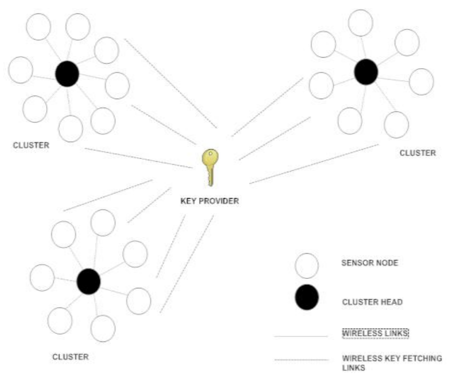

Figure 1: Basic Structure

The basic structure in figure 1 consists wireless sensor nodes with energies and chosen cluster head on basis of heapsort algorithm, all communicating with each other and key provider. Thus this protocol functions in the distributed manner using cluster heads and fulfills the requirements of distributed reprogramming protocol. The few of them are:

- Authenticity and integrity: The packets (code images) must be verified by sensor node before installation and only by trusted source.
- Distributed: The information is distributed with the help of different cluster heads. They keep interacting with key provider for updates.
- Time management: The communication-taking place here is in actual time intervals as shown in the graphs in the coming section for evaluation.
- Scalability: This protocol is highly scalable as well as energy efficient. It can accommodate higher number of nodes and thereby arranging them in heapsort order increases its efficiency and reduces energy overhead. Figure 2 discusses the flowchart of our proposed protocol.

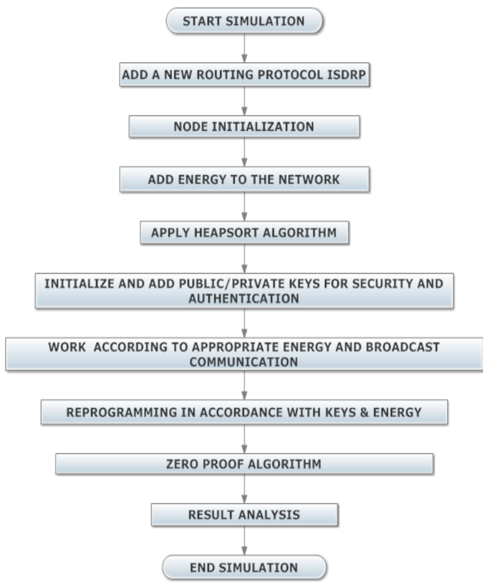
START SIMULATION
ADD A NEW ROUTING PROTOCOL ISDRP
NODE INITIALIZATION
ADD ENERGY TO THE NETWORK
APPLY HEAPSORT ALGORITHM
INITIALIZE AND ADD PUBLIC/PRIVATE KEYS FOR SECURITY AND AUTHENTICATION
WORK ACCORDING TO APPROPRIATE ENERGY AND BROADCAST COMMUNICATION
REPROGRAMMING IN ACCORDANCE WITH KEYS & ENERGY
ZERO PROOF ALGORITHM
RESULT ANALYSIS
END SIMULATION

Figure 2: Flowchart for ISDRP

We will explain our proposed work with the help of following scenarios. The figure 3 scenario depicts the position of initial position of the nodes at the starting point when simulation starts. Furthermore, we have to apply reprogramming procedure using requisite implementation to be carried out as per proposed work.

The figure 4 scenario clearly shows that the all nodes are ready to perform as per heap sort mechanism in which each node renders to give their information to the key provider (base station) and to get the relevant key for proper broadcasting or communication as per requisite in which there will be less amount of data loss as compare to before when they communicate in their cluster without information sharing to the key provider

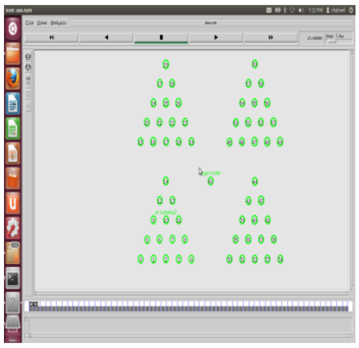
Figure 3: Initial Position of Nodes

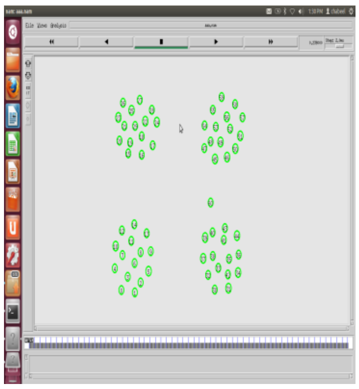
Figure 4: Heapsort Algorithm for Nodes

The figure 5 scenario clearly depicts the concept of information sharing of a node towards the key provider, after it they come back to their own place and again communicates and find out after this reprogramming of data through key information is highly applicable and less number of packets drop would be found then before. The figure 6 scenario depicts that whenever the role of all activities for proposed work gets completed and finishes to reprogram, the nodes gets to position and shape as per their initial start.

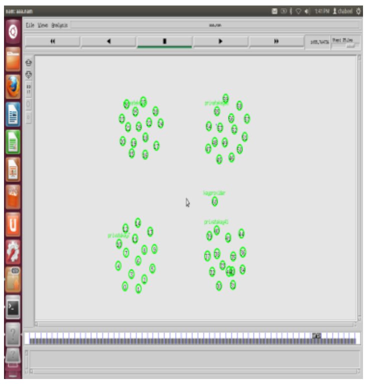
Figure 5: Nodes Sharing and Collecting Information

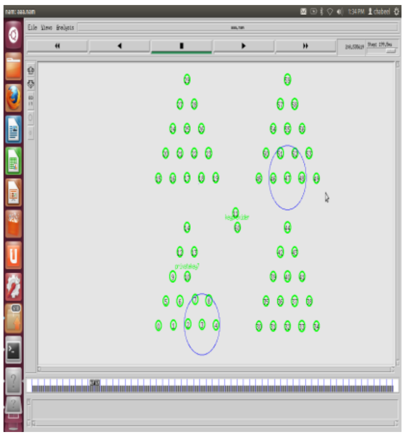

Figure 6: Repositioning of Nodes to Original Positions to and from Key Provider

IMPLEMENTATION AND PERFORMANCE EVALUATION

The implementation is done using network simulator ns 2.35 installed on the operating system Ubuntu 12.04 which a Linux based on laptop PC. The whole scenario consists of 59 nodes with one key provider making four clusters. Each cluster head is provided with private keys, which gets changed after some interval ensuring higher confidentiality using zero knowledge proof.

The throughput is being evaluated for the proposed protocol. It is the amount of packets delivered in a given time period in other words it is the average rate of successful message delivery over a communication channel. The metrics such as total send packets figure 7, receive packets figure 8, and drop packets figure 9 are considered. But overall throughput is the successful receive packets. The graph shows linear behavior with slight slope of increase at each interval because of the two values, one before is before getting public key from key provider and other is after getting the public key/private key.

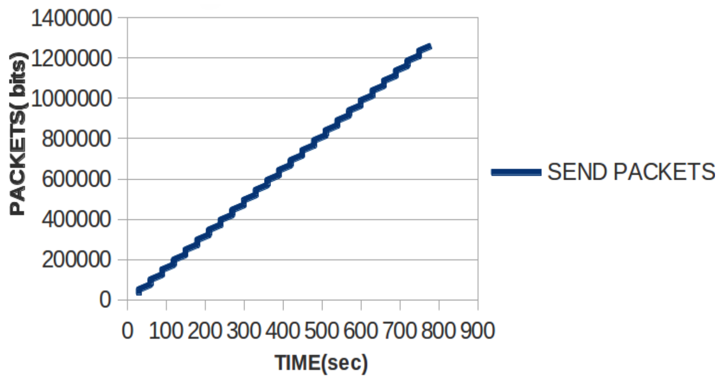

Figure 7: Total Number of Packets Sent

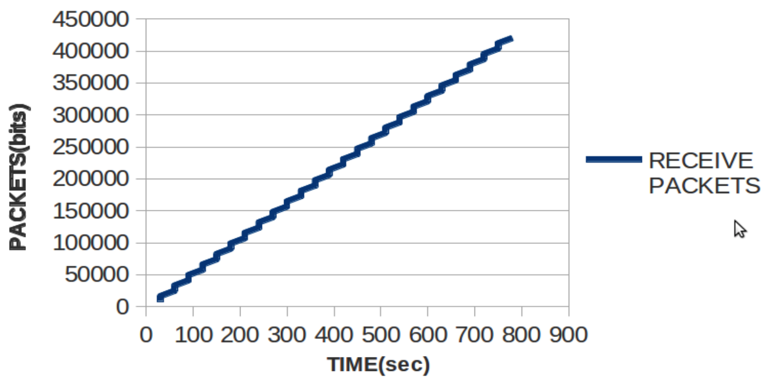

Figure 8: Total Number of Packets Received

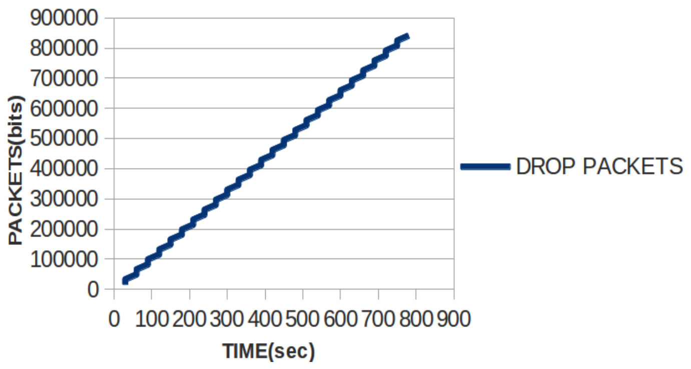

Figure 9: Total Packets Dropped

CONCLUSIONS

Nowadays, secure and distributed programming is a challenging problem from security point of view in which we should robustly replicate data for the requisite destination. The above proposed work tries to maintain the reality from security purpose in which we demonstrate the throughput result clearly with the help of send, receive and drop packets which is quiet reliable and robust with time.

Furthermore, the proposed routing protocol ISDRP is very effective as compare previous proposed routing protocols. Moreover, there is need to analyze the result more efficiently by using some more different parameters like network overhead, power consumption. There is somehow also need to increase life time validity by increasing network sequence number and forwarding data.

ACKNOWLEDGEMENTS

To the light, our god, who guided us through the way.To Mr. Hardeep Singh for his great efforts of supervising and leading us, to accomplish this fine work.To our friends and families, they were a great source of support and encouragement; we thank them all and wish them all the best in their lives.

REFERENCES

1. Boneh, Franklin Matthew. (2001). Identity-based encryption from the Weil pairing, Proceedings of Crypto 2001 of Lecture Notes in Computer Science, Vol. 2139, pp. 213–229.
2. Chlipala, Hui Jonathan., Tolle Gilman. (2003). Deluge: Data Dissemination for Network Reprogramming at Scale, University of California at Berkeley Computer Science Division, Berkeley.
3. Lanigan, Gandhi Rajeev., Narasimhan, Priya. (2006). Sluice: Secure dissemination of code updates in sensor networks, 26thInternational Conference on Distributed Computing Systems, pp.53.
4. Wang, Zhu Yaoyao., Cheng Liang. (2006). Reprogramming Wireless Sensor Networks: Challenges and Approaches, IEEE network, Vol.20, No.3, pp. 48-55.
5. Lipschutz, Pai G.A.V. (2006).Chapter7- Trees, Data Structures, Tata Mc Graw publishing company limited, New Delhi, pp. 7.1-7.101.
6. Varma, Tiwary U.S., Konakalla Ravikiran. (2007). Remote Reprogramming Mechanism for WSN, International Conference on Intelligent and Advanced Systems**,** Indian Institute of Information Technology, Allahabad, pp. 939-944.

7. Hyun, Ning Peng.,Liu An., and Du Wenliang. (2008). Seluge: Secure and DoS-Resistant Code Dissemination in Wireless Sensor Networks, International Conference on Information Processing in Sensor Networks, St. Louis, MO, pp. 445-456.

8. Wang, Balasingham Ilangko. (2010). Wireless Sensor Networks - An Introduction, Chapter 1 Wireless Sensor Networks: Application-Centric Design, pp. 1-13.

9. He, ChenChun., Chan Sammy., Bu Jiajun. (2012). SDRP: A Secure and Distributed Reprogramming Protocol for Wireless Sensor Networks, IEEE Transactions on Industrial Electronics, Vol. 59, No. 11, pp. 4155-4163.

10. Yadong, Lin Yao., Wengquan Wu., Xiaotong Zhang. (2011).A Hierarchical Online Reprogramming Method of Wireless Sensor Networks, 7th International Conference on Communications, Networking and Mobile Computing (WiCOM), Wuhan, pp. 1-4.

11. Zero knowledge proof [Online]. Available: http://en.wikipedia.org/wiki/Zero-knowledge_proof.

12. Mohr, A Survey of Zero Knowledge Proofs with Applications to Cryptography, Southern Illinois University, Carbondale, pp. 1-12.